

AMENDMENTS TO THE CLAIMS

1. (Original) A method for representing industrial automation computer program code created using a graphical programming language tool that stores the created code in computer memory in an internal representation during execution, the method comprising the steps of:
 identifying industrial automation code in computer memory in the internal representation;
 and
 converting the code from the internal representation to a markup language format.
2. (Original) The method according to claim 1, comprising the further step of causing the converted, markup-formatted code to be stored in a computer data storage device.
3. (Original) The method according to claim 1, further comprising the step of transmitting the markup-formatted code over a network to a receiving computing device.
4. (Original) The method according to claim 2, comprising the further steps of retrieving the markup-formatted code from the computer data storage device and converting the markup-formatted code to the internal representation in computer memory.
5. (Original) The method according to claim 2, comprising the further steps of retrieving the markup-formatted code from the computer data storage device and representing the retrieved code in a corresponding graphic format on a computer display.
6. (Original) The method according to claim 5, wherein the display of the markup-formatted code is facilitated by a browser.

7. (Original) The method according to claim 2, wherein the markup language is XML.
8. (Original) The method according to claim 1, wherein the graphical programming language comprises a flowchart language.
9. (Original) The method according to claim 1, wherein the graphical programming language comprises a ladder logic language.
10. (Original) The method according to claim 1, wherein the graphical programming language comprises a function block diagram language.
11. (Original) The method according to claim 1, wherein the graphical programming language comprises a sequential function chart.
12. (Original) The method according to claim 7, wherein the graphical programming language comprises a flowchart language.
13. (Original) The method according to claim 7, wherein the graphical programming language comprises a ladder logic language.
14. (Original) The method according to claim 7, wherein the graphical programming language comprises a sequential function chart.
15. (Original) The method according to claim 7, wherein the graphical programming language comprises a sequential function block diagram language.

16. (Original) The method according to claim 1, wherein the graphical programming language tool comprises an editor and the conversion is triggered by invoking an editor command.

17. (Original) The method according to claim 7, comprising the further steps of retrieving the markup-formatted code from the computer data storage device and representing the retrieved code in a corresponding graphic format on a computer display.

18. (Original) The method according to claim 17, wherein the step of displaying the code on a computer display device comprises the step of displaying the code with the use of a browser.

19. (Original) A computer program product, for use in conjunction with a computing device, for creating industrial automation system control program code using a graphical language programming tool and storing the code in a computer memory in an internal representation during execution, the computer program product comprising a computer usable medium, the computer usable medium comprising:

computer readable program code for identifying industrial automation control program code stored in computer memory in the internal representation;

computer readable program code for converting the identified industrial automation control program code from the internal representation to a markup language format.

20. (Original) The computer program product according to claim 19, the computer usable medium further comprising computer readable program code for causing the converted, markup-formatted code to be stored in a computer data storage device.

21. (Original) The computer program product according to claim 20, the computer usable

medium further comprising computer readable program code for causing retrieval of the markup-formatted code from the computer data storage device and converting the markup-formatted code to the internal representation in computer memory.

22. (Original) The computer program product according to claim 19, the computer usable medium further comprising computer readable program code for causing the transmission of markup-formatted code over a network to a receiving computing device.

23. (Original) The computer program product according to claim 20, the computer program product further comprising computer readable program code for retrieving the markup-formatted code from the computer data storage device and representing the retrieved code in a corresponding graphic format on a computer display.

24. (Original) The computer program product according to claim 23, wherein the display of the markup-formatted code is facilitated by a browser.

25. (Original) The computer program product according to claim 19, wherein the markup language is XML.

26. (Original) The computer program product according to claim 19, wherein the graphical programming language comprises a flowchart language.

27. (Original) The computer program product according to claim 19, wherein the graphical programming language comprises ladder logic.

28. (Original) The computer program product according to claim 19, wherein the graphical

programming language comprises function block diagrams.

29. (Original) The computer program product according to claim 19, wherein the graphical programming language comprises a sequential function chart.

30. (Original) The computer program product according to claim 25, wherein the graphical programming language comprises a flowchart language.

31. (Original) The computer program product according to claim 25, wherein the graphical programming language comprises a ladder logic language.

32. (Original) The computer program product according to claim 25, wherein the graphical programming language comprises a function block diagram language.

33. (Original) The computer program product according to claim 25, wherein the graphical programming language comprises a sequential function chart.

34. (Original) The computer program product according to claim 19, further comprising computer readable program code for converting the markup-formatted code to the graphical programming language internal representation.

35. (Original) The computer program product according to claim 19, wherein the computer program product graphical language programming tool comprises an editor, and wherein the conversion is triggered by invoking an editor command in the graphical programming language editor.

36. (Original) A computer program product comprising a computer-readable storage medium and having data stored thereon, the data comprising a representation of industrial automation control code formatted in a markup language.

37. (Original) The computer program product according to claim 36, wherein the markup language is XML.

38. (Original) The computer program product according to claim 36, wherein the computer program product is coupled to a computing system that is remotely located from an industrial automation control system.

39. (Original) A computer program product for permitting a user to create industrial automation control programs, the product comprising a computer-readable storage medium having computer program code stored on it, the code comprising:

industrial automation graphical programming language code, the graphical programming language code comprising an editor adapted to permit the user to create industrial automation control code using graphical elements, the control code being stored in memory in an internal representation during execution; and

computer program code for converting industrial automation control code, stored in memory in the internal representation, from the internal representation to a markup language format.

40. (Original) The computer program product according to claim 39, further comprising computer program code for converting industrial automation control code from the markup language format to the internal representation.

41. (Original) A method for communicating the logical structure of industrial automation control program data in order to permit a plurality of application developers to create applications relating to the data, the method comprising the steps of:

creating a schema defining a content model for markup language files generated by an industrial automation control program system; and

posting the schema for access over a network by the application developers.

42. (Original) The method according to claim 41, wherein the schema is an XML schema.

43. (Original) The method according to claim 41, wherein the industrial automation control program data comprises flowchart programming instructions.

44. (Original) A method for providing industrial automation control code from a server system, over a network to which the server system is coupled, and to a client system also coupled to the network, the method comprising the steps of:

accessing a markup-formatted version of the control code;

transmitting the accessed, markup-formatted control code over the network in connection with a network address corresponding to the client system, thereby causing the transmitted, markup-formatted control code to be received by the client system.

45. (Currently Amended) The method according to claim 44, wherein the client devicesystem, in response to the received markup-formatted control code, has transmitted to the server system data relating to the automation to which the markup-formatted control code is directed, and, further, wherein the server system has access to control code modified in response to receipt of the data from the client system, and wherein the modified control code is markup-formatted, the method comprising the further step of:

transmitting the markup-formatted, modified control code over the network in connection with a network address corresponding to the client system, thereby causing the transmitted, modified, markup-formatted control code to be received by the client system.

46. (Original) The method according to claim 45, wherein the step of transmitting the accessed, markup-formatted control code over the network comprises sending an electronic mail message.

47. (Original) The method according to claim 45, wherein the step of transmitting the accessed, markup-formatted control code over the network comprises transmitting the code over the network via hypertext transfer protocol.

48. (Original) The method according to claim 44, wherein the markup-format of the control code comprises XML.

49. (Original) The method according to claim 44, wherein a second client system is coupled to the network, the method further comprising the step of:

transmitting the accessed, markup-formatted control code over the network in connection with a network address corresponding to the second client system, thereby causing the transmitted, markup-formatted control code to be received by the second client system.

50. (Original) The method according to claim 49, wherein the first client system is configured to reconvert the markup-formatted control code to a first internal representation, and wherein the second client system is coupled to the network, the second client configured to reconvert the markup-formatted control code to a second internal representation.

51. (Original) A method for programming industrial automation control applications comprising the steps of:

- providing a computer system coupled to a network;
- configuring the first computer system to receive over the network transmissions of data from a plurality of industrial automation program developer systems; and
- receiving data from the plurality of industrial automation program developer systems program code in a markup language format.

52. (Original) The method according to claim 51, wherein the markup language is XML.